Section 1.7 : The ~~basic~~ types $0, 1, \cancel{2}$

Defin. 1.7.1 The empty type 0.

Form_0 : $0 : \mathcal{U}$

Intro_0 : no point-constructors for 0

Rec_0 : If $A : \mathcal{U}$, there is $\mathrm{Efq}_A : 0 \to A$ without any defining equations

i.e., the type $0 \to A$ is always inhabited by the object Efq_A (it corresponds to the logical rule: Ex falso quodlibet)

Ind_0 : Let $P : 0 \to \mathcal{U}$ (e.g. $\mathrm{Efq}_{\mathcal{U}} : 0 \to \mathcal{U}$, i.e., such type families exist by Rec_0, otherwise with the help of some trivial judgement)

There is $F_P : \prod_{x:0} P(x)$ without giving any defining eq. for F_P

Remark 1.7.2 : Let $P : 0 \to \mathcal{U}$ be the constant type family A. Then

$$F_{\overline{A}_0} : \prod_{x:0} A \equiv 0 \to A$$

i.e. the $\mathrm{Ind}_0 \Rightarrow \mathrm{Rec}_0$.

This is going to be always the case: $\mathrm{Rec}_{\mathrm{Type}} \xLeftarrow{\text{(constant type fams)}} \mathrm{Ind}_{\mathrm{Type}}$.

It is useful though to have $\mathrm{Rec}_{\mathrm{Type}}$ for every inductively defined type given separately, for emphasis. (It would suffice of course to mention only Ind (Exercise) here)

Definition 1.7.3 : If $A : \mathcal{U}$, we define $\neg A \equiv A \to 0$, the negation of the type (proposition) A.

- A is inhabited $:\Leftrightarrow$ there is $a : A$
- A not inhabited $\Leftrightarrow$ $\neg A$ is inhabited
  $\Leftrightarrow a : A \to 0$, i.e. $a(u) : 0$ for every supposed $u : A$.

Corollary 1.7.4 : 0 is not inhabited

Proof : $\mathrm{Efq}_0 : 0 \to 0$ i.e. $\neg 0$ is inhabited, by Rec_0.

Corollary 1.7.5 $\prod_{x:0} \left(Efq_0(x) =_0 id_0(x) \right)$ is inhabited

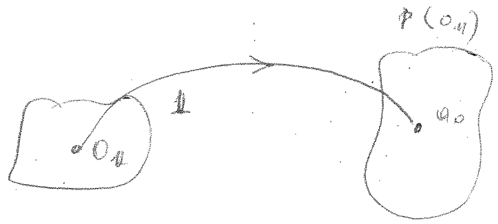
Proof: $P : 0 \to U \qquad P(x) :\equiv \left(Efq_0(x) =_0 id_0(x) \right)$

$F_P : \prod_{x:0} P(x)$ by Ind_0.

Remark 1.7.6:
Note that such results are of no importance: If $P : 0 \to U$ and $\neg P : 0 \to U$ $(\neg P)(x) \equiv \neg P(x)$ for $x : 0$ then $F_P : \prod_{x:0} P(x)$ and $F_{\neg P} : \prod_{x:0} (\neg P)(x)$, but since there is no $x : 0$, we can't have a contradiction.

Remark 1.7.7 Consistency of ITT. Following the rules of ITT, we cannot obtain an element of 0.
This metatheoretic claim is shown in HoTT-book p.548.

Definition 1.7.8: Canonical type $\mathbb{1}$.



$\text{Form}_{\mathbb{1}}$: $\mathbb{1} : U$

$\text{Intro}_{\mathbb{1}}$: $\dfrac{}{0_{\mathbb{1}} : \mathbb{1}}$

$\text{Rec}_{\mathbb{1}}$: $\dfrac{A : U \quad a : A}{f : \mathbb{1} \to A \text{ st. } f(0_1) \equiv a}$

$\text{Ind}_{\mathbb{1}}$: $\dfrac{P : \mathbb{1} \to U \quad a_0 : P(0_{\mathbb{1}})}{F : \prod_{x:\mathbb{1}} P(x) \text{ st. } F(0_{\mathbb{1}}) \equiv a_0}$

- If a "copy" of the introduced type is given somewhere in some other type, then the defined is defined onto this structure. (dependent + non-dependent version)
- This pattern is followed ✓.
- It corresponds to being the "least" with the property
- Clearly $Ind_1 \Rightarrow Rec_1$
- it suffices to find $a_0 : P(0_1)$ to find F. (as expected, one inhabitant determines F).

Proposition 1.7.9 (Uniqueness Principle for $\mathbb{1}$) The TFTII (Following type is inhabited)
(Ex?)

$$\prod_{x:\mathbb{1}} \left(x =_{\mathbb{1}} 0_{\mathbb{1}} \right)$$

Remark 1.7.10
every point equal to the output ~~[illegible]~~ point-constructor. (and not the output of some point-constructor)
UP(2) more delicate

Proof: $P : \mathbb{1} \to U \qquad P(x) \equiv (x =_1 0_1)$

$P(0_1) \equiv (0_1 = 0_1) \qquad refl_{0_1} : (0_1 = 0_1)$

By Ind_1 there is $F : \prod_{x:1} (x =_1 0_1)$ st $F(0_1) \equiv refl_{0_1}$.

Section 1.8. Product Types

Definition 1.8.1

- Form$_\times$:
$$\frac{A, B : \mathcal{U}}{A \times B : \mathcal{U}}$$

- Intro$_\times$:
$$\frac{x : A \ , \ y : B}{(x, y) : A \times B}$$

- Rec$_\times$:
$$\frac{C : \mathcal{U} \ , \ g : A \to B \to C}{f : A \times B \to C \quad \text{s.t.} \quad f((x,y)) \equiv g(x,y) \equiv g(x)(y) \ , \text{ for every } x : A, y : B}$$

- Ind$_\times$:
$$\frac{P : A \times B \to \mathcal{U} \qquad G : \prod_{x:A} \prod_{y:B} P((x,y))}{F : \prod_{z : A \times B} P(z) \quad \text{s.t.} \quad F((x,y)) \equiv G(x,y) \quad \text{for every } x : A, y : B}$$

Proposition 1.8.2 There are functions $pr_{1,\times} : A \times B \to A$, $pr_{2,\times} : A \times B \to B$ such that for every $x : A$, $y : B$ we have:

$$pr_{1,\times}((x,y)) \equiv x \quad , \quad pr_{2,\times}((x,y)) \equiv y$$

Proof: By Rec$_\times$ if $g_1 : A \to B \to \overset{C}{\overset{\shortparallel}{A}}$, $g_2 : A \to B \to \overset{C}{\overset{\shortparallel}{B}}$ are defined by

$$g_1(x,y) \equiv x \quad , \quad g_2(x,y) \equiv y \quad , \text{ for every } x : A, y : B,$$

there are
$$f_1 : A \times B \to A \quad \text{s.t.} \quad f_1((x,y)) \equiv g_1(x,y) \equiv x$$
$$f_2 : A \times B \to B \quad \text{s.t.} \quad f_2((x,y)) \equiv g_2(x,y) \equiv y.$$

$$f_1 \equiv pr_{1,\times} \quad , \quad f_2 \equiv pr_{2,\times}$$

Proposition 1.8.3 Uniqueness Principle for product types (Uniq. Pr. for $\times$) TFTH:
$$\prod_{z : A \times B} \left(z =_{A \times B} (pr_{1,\times}(z), pr_{2,\times}(z)) \right)$$

Proof: We use necessarily Ind$_\times$.

Let $P : A \times B \to \mathcal{U}$ defined by
$$P(z) \equiv \left(z =_{A \times B} (pr_{1,\times}(z), pr_{2,\times}(z)) \right)$$

We need to define $G : \prod_{x:A} \prod_{y:B} \left((x,y) =_{A\times B} (pr_{1,\times}((x,y)), pr_{2,\times}((x,y))) \right)$

i.e., $G(x,y) : (x,y) =_{A\times B} (pr_{1,\times}((x,y)), pr_{2,\times}((x,y)))$

$$\parallel \ (\text{Prop. } 1.8.2)$$

$$(x,y) =_{A\times B} (x, y)$$

define $G(x,y) :\equiv refl_{(x,y)}$

By $Ind_\times$: $F : \prod_{z:A\times B} P(z)$ st $\underline{F((x,y)) \equiv refl_{(x,y)}}$. ∎

• **Corollary 1.8.4**: For every $z : A\times B$ there exist $a : A$ and $b : B$ st $z =_{A\times B} (a,b)$.

<u>Proof</u>: Apply Prop. 1.8.3, since $pr_{1,\times}(z) : A$ and $pr_{2,\times}(z) : B$ ∎

②

Exercise: Conversely, the recursor for product types $rec_{A\times B}$ can be derived from the projection maps pr_1, pr_2

Proof: (a) In the context of ①.

let $g : A \to B \to C$

and $pr_1 : A\times B \to A$ $\quad pr_2 : A\times B \to B$

$pr_1((a,b)) = a$ $\quad pr_2((a,b)) = b$

$$f((a,b)) = g(pr_1(a,b))(pr_2(a,b))$$

$$= g(a)(b)$$

~~$f = g$~~

(b) In the context of ② (school solution)

$$rec_{A\times B}(C, g, (a,b)) = g(pr_1(a,b))(pr_2(a,b))$$

Ex

Ex

Let $pr_{1,x} : A\times B \to A$ $\quad pr_{2,x} : A\times B \to B$ st.

$pr_{1,x}((x,y)) \equiv x$ $\quad pr_{2,x}((x,y)) \equiv y$

then If $g : A \to B \to C$, there is $f : A\times B \to C$ st. $f((x,y)) = g(x,y)$

Proof: $f(z) = g(pr_1(z), pr_2(z))$

Hence $f((x,y)) = g(x,y)$

§ Logical equivalence of types

- $A \leftrightarrow B :\equiv (A \to B) \times (B \to A)$ Ex (types A, B are logically equivalent)

Example $\mathbb{N} \leftrightarrow \mathbb{1} :\equiv (\mathbb{N} \to \mathbb{1}) \times (\mathbb{1} \to \mathbb{N})$ $\mathbb{1} \leftrightarrow \mathbb{1}$

We need to find $x : (\mathbb{N} \to \mathbb{1}) \times (\mathbb{1} \to \mathbb{N})$

$a : \mathbb{N} \to \mathbb{1}$ $\qquad$ $b : \mathbb{1} \to \mathbb{N}$

$a = \lambda n. *$ $\qquad$ $b(*) = 0$

$(a, b) : (\mathbb{N} \to \mathbb{1}) \times (\mathbb{1} \to \mathbb{N})$

This trivial fact shows the unimportance of $\leftrightarrow$ and shows that $\simeq$ of types captures the essence of a type!

- But $\mathbb{N}, \mathbb{1}$ are not equivalent as types $\left(A \simeq B :\equiv \sum_{f : A \to B} \text{isequiv}(f) \right)$ (p. 100).

$$\text{isequiv}(f) :\equiv \left(\sum_{g : B \to A} f \circ g \sim id_B \right) \times \left(\sum_{h : B \to A} h \circ f \sim id_A \right)$$

defined on function types

explain exactly why $\mathbb{N}, \mathbb{1}$ are not equivalent as types (?)

intuitively: $\mathbb{N}$ has ω-pieces, $\mathbb{1}$ only one.
$\mathbb{N}$ is not path connected, $\mathbb{1}$ is path-connected

not expected

- equivalence of types $\equiv$ logical equivalence for mere props

6α.

- $A \to \neg\neg A$

$A \to ((A \to 0) \to 0)$ is inhabited

$x : A$

$f : A \to 0$ i.e. $f(x) : 0$

Define $g \equiv \lambda(x : A) . \big(\lambda (f : A \to 0) . f(x) \big)$

i.e. $g(x, f) = f(x)$.

- $\neg\neg A \to A$ cannot be inhabited

$((A \to 0) \to 0) \to 0$

$\lambda \big(f : ((A \to 0) \to 0) \big) . (\quad)$

$u : A \to 0$

$f(u) : 0$

$a : A$, $u(a)$

immediate

no way to define $a : A$ from these data

- (Brouwer) $\boxed{\neg\neg\neg A \to \neg A \text{ is inhabited}}$

$\big[((A \to 0) \to 0) \to 0 \big] \to (A \to 0)$

Από την $F : ((A \to 0) \to 0) \to 0$

πρέπει να φτιάξω element του $A \to 0$

$u : A \to 0$

$u(a)$

$f : (A \to 0) \to 0$

$F(f) : 0$

let $a : A \overset{F}{\rightsquigarrow} u(a) : 0$

από το a θα φτιάξω $\hat{a} : (A \to 0) \to 0$

$\hat{a}(f) \equiv f(a)$ hence $F(\hat{a}) : 0$

$$\boxed{u \equiv \lambda(a : A) . F(\hat{a}) \quad : A \to 0}$$

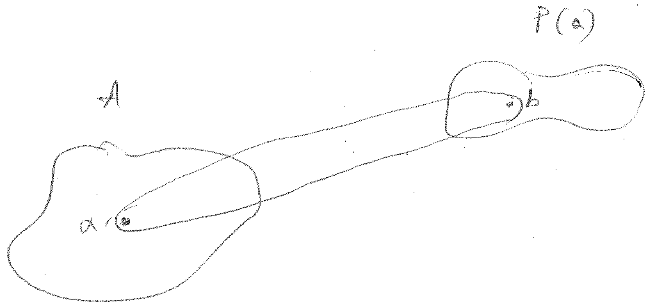
$\square$

Section 1.5 Dependent pair types (Σ-types)

- Codependent version of product types, as $\prod_{x:A} P(x)$ of $A \to B$)
- (It corresponds to $\exists x \in A\ (P(x))$)

Definition 1.5.1:

Form$_\Sigma$:
$$\frac{A : \mathcal{U} \qquad P : A \to \mathcal{U}}{\sum_{x:A} P(x) : \mathcal{U}}$$



Intro$_\Sigma$:
$$\frac{a : A \qquad b : P(a)}{(a,b) : \sum_{x:A} P(x)}$$

- Rec$_\Sigma$:
$$\frac{C : \mathcal{U} \qquad G : \prod_{x:A} (P(x) \to C)}{f : \left(\sum_{x:A} P(x)\right) \to C \quad \text{s.t.} \quad f((x,u)) \equiv G(x,u), \text{ for every } x:A,\ u:P(x)}$$

- Ind$_\Sigma$:
$$\frac{\begin{array}{c} Q : \left(\sum_{x:A} P(x)\right) \to \mathcal{U} \\ G : \prod_{x:A} \prod_{u:P(x)} Q((x,u)) \end{array}}{F : \prod_{z : \sum_{x:A} P(x)} Q(z) \quad \text{s.t.} \quad F((x,u)) \equiv G(x,u), \text{ for every } x:A,\ u:P(x)}$$

i.e., a (dependent) function of (x,u) is determined by functions of $(x)(u)$.

Remark 1.5.1 If $P = \overline{B}_A$, then $\sum_{x:A} P(x) \equiv \sum_{x:A} B$, then $G : \prod_{x:A} (B \to C) \equiv A \to (B \to C)$ (in Rec$_\Sigma$)
$\equiv A \to B \to C$

$F : \left(\sum_{x:A} B\right) \to C \quad F((x,b)) \equiv G(x,b)$ is Rec$_\times$

Similarly for Ind$_\times$

We can package the recursion principle $\mathrm{Rec}_{\sum}$ and the induction principle $\mathrm{Ind}_{\sum}$ into a single function, respectively, as follows:

$$\mathrm{Rec}_{\sum} :$$

with the defining equation

$$,$$

$$\mathrm{Ind}_{\sum} :$$

with the defining equation

$$.$$

Proposition 1.8.1. *The first projection function* $\mathrm{pr}_{1,\sum} : (\sum_{x:A} P(x)) \to A$ *where* $\mathrm{pr}_{1,\sum}((x,p)) :\equiv x$, *for every* $x : A$ *and* $p : P(x)$, *and the second projection function*

$$\mathrm{pr}_{2,\sum} : \prod_{q:\sum_{x:A} P(x)} P(\mathrm{pr}_{1,\sum}(q)),$$

where $\mathrm{pr}_{2,\sum}((x,p)) :\equiv p$, *for every* $x : A$ *and* $p : P(x)$, *are definable.*

Ex.1

Proof. With the use of $\mathrm{Rec}_{\sum}$, if $G : \prod_{x:A}(P(x) \to A)$ is defined by

$$G(x,p) :\equiv x,$$

for every $x : A$, the first projection function

$$\mathrm{pr}_{1,\sum} : \left(\sum_{x:A} P(x)\right) \to A$$

is defined by

$$\mathrm{pr}_{1,\sum}((x,p)) :\equiv G(x,p) \equiv x,$$

for every $x : A$ and $p : P(x)$.

With the use of $\mathrm{Ind}_{\sum}$, if $Q : (\sum_{x:A} P(x)) \to \mathcal{U}$ is defined by

$$Q(q) :\equiv P(\mathrm{pr}_{1,\sum}(q)),$$

for every $q : \sum_{x:A} P(x)$, and if $G : \prod_{x:A} \prod_{p:P(x)} P(\mathrm{pr}_{1,\sum}(q))$ is defined by

$$G(x,p) :\equiv p,$$

for every $x : A$ and $p : P(x)$, then the second projection function

$$\mathrm{pr}_{2,\sum} : \prod_{q:\sum_{x:A} P(x)} P(\mathrm{pr}_{1,\sum}(q))$$

is defined by

$$\mathrm{pr}_{2,\sum}((x,p)) :\equiv G(x,p) \equiv p,$$

for every $x : A$ and $p : P(x)$. ∎

Cells here on the $p : P(x)$, $u = (a, p)$

Uniqueness: $\prod_{u:\sum_{x:A} P(x)} \left(u =_{\sum_{x:A} P(x)} (\mathrm{pr}_{1,\sum}(u), \mathrm{pr}_{2,\sum}(u))\right)$ as in the $\underline{x\text{-case}}$

<u>Proposition 1.1.2</u>: There are functions $pr_1 : \left(\sum_{x:A} P(x)\right) \to A$, ~~such that~~

$$pr_2 : \prod_{z:\sum_{x:A} P(x)} P(pr_1(z))$$

such that

$$pr_1((x,u)) \equiv x \quad \text{for every } x:A,\ u:P(x)$$

$$pr_2((x,u)) \equiv u, \quad \text{—//—}$$

<u>Proof</u>: (Exercise)

<u>Exercise</u>: Formulate $UP(\Sigma)$ and prove it

<u>Proposition (1.1.3)</u>: (Uniqueness principle for Σ-types) ~~For Today~~:

$$\prod_{z:\sum_{x:A} P(x)} \left(z =_{\sum_{x:A} P(x)} (pr_1(z), pr_2(z))\right)$$

<u>Proof</u>: Let $Q : \left(\sum_{x:A} P(x)\right) \to U$ be defined by $Q(z) \equiv \left(z =_{\sum_{x:A} P(x)} (pr_1(z), pr_2(z))\right)$

We need to find $G : \prod_{x:A} \prod_{u:P(x)} Q((x,u)) \equiv \prod_{x:A} \prod_{u:P(x)} (x,u) =_{\sum_{x:A} P(x)} (pr_1((x,u)), pr_2((x,u))$

$$\equiv \prod_{x:A} \prod_{u:P(x)} \left((x,u) =_{\sum_{x:A} P(x)} (x,u)\right)$$

define $\quad G \equiv \lambda(x:A, u:P(x)).\ refl_{(x,u)}$

By ind_Σ there is $F : \prod_{z:\sum_{x:A} P(x)} Q(z)$ st $F((x,u)) \equiv G(x,u) \equiv refl_{(x,u)}$

F is the required inhabitant $\square$

<u>Corollary 1.1.4</u> For any $z:\sum_{x:A} P(x)$ there are $x:A$, $u:P(x)$ st $\left(z =_{\sum_{x:A} P(x)} (x,u)\right)$

<u>Proof</u>: Take $x \equiv pr_1(z)$, $u \equiv pr_2(z)$ and use Prop 1.1.3.

$A \xrightarrow{f} B \xrightarrow{g} A$

If $g \circ f = id_A$, then f is a right inverse of g.

- If $A \xrightarrow{f} B$ has a right inverse it maps A onto B.

$g : B \xrightarrow{g} A \xrightarrow{f} B$ $\qquad f \circ g = id_B$.

$(b \in B \to f(g(b)) = b)$

- $\exists g : B \to A$ s.t. $f(g(b)) = b$, $\forall b \in B$

operation (not necessarily $g(b_1) = g(b_2)$, for $b_1 = b_2$)

How is it connected to Diaconescu?

Ex. 2 p. 58: Construct a surjection which is not onto.

(This changes in BBPT see p. 64 here onto = surjection)

§ 6.7

A ΝΑΦΟΡΑ of general notion of subtype?

$$\sum_{x : A} P(x)$$

$$\{x : A \mid P(x)\}$$

(In HoTT - basic motto when each P(x) is a proposition)

2

~~Section 11.2~~ The uniqueness principle for the equality type $x =_A y$.

~~Proposition 11.9.5~~ UP(=) Let $A : \mathcal{U}$, $a : A$

$$E_a \equiv \sum_{x : A} (a =_A x)$$

Then is $$\Lambda_a : \prod_{u : E_a} \left((a, \text{refl}_a) =_{E_a} u \right)$$

s.th. $$\Lambda_a\left(\left(a, \text{refl}_a \right)\right) \equiv \text{refl}_{(a, \text{refl}_a)}$$

Proof: (with the use of based path induction) Since E_a is a Σ-type we use Ind_Σ.

Let $P : E_a \to \mathcal{U}$ defined by $P(u) \equiv ((a, \text{refl}_a) =_{E_a} u)$

For Ind_Σ we need to find $G : \prod_{x : A} \prod_{p : a =_A x} P((x, p)) \equiv \prod_{x : A} \prod_{p : a =_A x} ((a, \text{refl}_a) =_{E_a} (x, p))$

Since this G is the output of based path ind (as c type) we define

$$C : \prod_{x : A} \prod_{p : a =_A x} \mathcal{U} \quad \text{by} \quad C(x, p) \equiv \left((a, \text{refl}_a) =_{E_a} (x, p) \right)$$

$$C(a, \text{refl}_a) \equiv \left((a, \text{refl}_a) =_{E_a} (a, \text{refl}_a) \right)$$

Take $c \equiv \text{refl}_{(a, \text{refl}_a)} : C(a, \text{refl}_a)$

Then there is G as above st $G(a, \text{refl}_a) \equiv \text{refl}_{(a, \text{refl}_a)}$

Clearly, Λ_a is the result of Ind_Σ on G

i.e. $\Lambda_a : \prod_{u : E_a} P(u)$ st $\Lambda_a\left((a, \text{refl}_a)\right) \equiv G(a, \text{refl}_a) \equiv$

$\equiv \text{refl}_{(a, \text{refl}_a)}$

∎

Theorem 1.26 : $\boxed{UP(=) \text{ and based-Transport} \Rightarrow \text{based path-induction}}$

Proof: Let $A : \mathcal{U}$, $a : A$

$$C : \prod_{x:A} \prod_{p : a =_A x} \mathcal{U} \quad \text{and} \quad c : C(a, \text{refl}_a)$$

We need to find $F : \prod_{x:A} \prod_{p : a =_A x} C(x,p)$ s.t. $F(a, \text{refl}_a) \equiv c$

Let Λ_a given by $UP(=)$

Let $P : E_a \to \mathcal{U}$ st $P((x,p)) \equiv C(x,p)$

$\boxed{\text{By } Rec_{\Sigma} \text{ for } \mathcal{U} \equiv C \text{ and } \mathcal{U}\equiv\mathcal{U}'}$ $C \equiv \mathcal{U} : \mathcal{U}'$ $G : \prod_{x:A} ((a =_A x) \to \mathcal{U})$

$G(x,p) \equiv C(x,p)$

then there is $P : E_a \to \mathcal{U}$ st $P((x,p)) \equiv C(x,p)$

Since $\Lambda_a((x,p)) : \left((a, \text{refl}_a) =_{E_a} (x,p)\right)$

$$[\Lambda_a((x,p))]_*^P : P((a,\text{refl}_a)) \to P((x,p))$$
$$\equiv$$
$$C(a, \text{refl}_a) \to C(x,p)$$

Hence we define

$$F \equiv \lambda(x : A, p : a =_A x). \; [\Lambda_a((x,p))]_*^P (c)$$

and

$$F(a, \text{refl}_a) \equiv [\Lambda_a((a, \text{refl}_a))]_*^P (c)$$
$$\equiv (\text{refl}_{(a,\text{refl}_a)})_*^P (c)$$
$$\equiv \text{id}_{P((a,\text{refl}_a))} (c)$$
$$\equiv \text{id}_{C(a,\text{refl}_a)} (c)$$
$$\equiv c \quad \text{since } c : C(a, \text{refl}_a)$$

$\square$

TYPE-THEORETIC justification of path-ind. 1.11 (Important to Program practical libraries).

<u>Theorem 19.7</u> Let $A, B : \mathcal{U}$ and $R : A \to B \to \mathcal{U}$ a relation on A, B be given. There is a function: (axiom of choice)

$$ac : \left(\prod_{x:A} \sum_{y:B} R(x,y)\right) \to \left(\sum_{f:A\to B} \prod_{x:A} R(x, f(x))\right)$$

$$\forall_{x\in A} \exists_{y\in B} (R(x,y)) \to \exists_{f:A\to B} \forall_{x\in A} (R(x, f(x)))$$

• expected to hold in Constr. math by the int. interpretation of $\forall, \exists$ (BHK).

<u>Proof:</u> Let $G : \prod_{x:A} \sum_{y:B} R(x,y)$ i.e.

$$G(x) : \sum_{y:B} R(x,y) \quad \text{i.e.,}$$

By Prop. 19.3. ($\Uparrow$ Ind_Σ).

$$\underset{\sum_{y:B} R(x,y)}{G(x) =} (pr_1(G(x)), pr_2(G(x))), \text{ where } pr_1(G(x)) : B \text{ and}$$
$$\bullet\ pr_2(G(x)) : R(x, pr_1(G(x)))$$

By these data we want to define a function $f : A \to B$

Clearly $f \equiv \lambda(x:A).\, pr_1(G(x))$ is such a function.

For this f we check if there is an element $F : \prod_{x:A} R(x, f(x)) \equiv \prod_{x:A} R(x, pr_1(G(x)))$

Clearly $F \equiv \lambda(x:A).\, pr_2(G(x))$

Hence we define $ac \equiv \lambda\left(G : \prod_{x:A} \sum_{y:B} R(x,y)\right).\,(f, F)$

i.e., $ac(G) \equiv (f, F)$

• <u>Remark 19.8.</u> What we actually did was the following:

① Unfold the given data (no deep, creative thinking), mechanical procedure (we only used Ind_Σ) and the explicit def of function $A \to B$, $\prod_{x:A} P(x)$

② We saw that: the data suffice / there is a solution

(First) instance of Naturality Thesis.

$f : A \to B$

Remark 1.1.9. (exercise)

$$\prod_{x:A} \sum_{y:B} \big(\underbrace{f(x) =_B y}_{R(x,y)} \big) \quad \text{is inhabited by} \quad F \equiv \lambda(x:A).\ (f(x), \mathrm{refl}_{f(x)})$$

By AC $\quad ac(F) : \sum_{g : A \to B} \prod_{x:A} (f(x) = g(x))$ $\qquad ac(F) \equiv (g, G)$ s.t.

$$\begin{aligned} g &\equiv \lambda(x:A).\ pr_1(F(x)) \\ &\equiv \lambda(x:A).\ pr_1(f(x), \mathrm{refl}_{f(x)}) \\ &\equiv \lambda(x:A).\ f(x) \\ &\equiv f. \end{aligned}$$

i.e., we get f again (Expected)

$$\mathrm{ind}_{A+B} : \prod_{C:(A+B)\to \mathcal{U}} \left(\prod_{a:A} C(\mathrm{inl}(a)) \right) \to \left(\prod_{b:B} C(\mathrm{inr}(b)) \right) \to$$

$$\to \prod_{x:A+B} C(x)$$

s.t.

$$\mathrm{ind}_{A+B}(C, g_0, g_1, \mathrm{inl}(a)) = g_0(a)$$

$$\mathrm{ind}_{A+B}(C, g_0, g_1, \mathrm{inr}(a)) = g_1(b)$$

(when) a in all other cases

Exercise

If $x : A+B$, then there exist $a : A$ ~~b : B~~ s.t.

$$x =_{A+B} \mathrm{inl}(a) \quad \text{or}$$

there exists $b : B$ st

$$x =_{A+B} \mathrm{inr}(b)$$

$$\prod_{z:A+B} \Bigg[\underbrace{\left(\sum_{x:A} (z = \mathrm{inl}(x)) \right) + \left(\sum_{y:B} z = \mathrm{inr}(y) \right)}_{P(z)} \Bigg]$$

$$G_\ell : \prod_{x:A} P(\mathrm{inl}(x)) = \prod_{x:A} \left(\sum_{x:A} (\mathrm{inl}(x) = \mathrm{inl}(x)) + \sum \right)$$

$$G_\ell(x) = \mathrm{inl}(x, \mathrm{refl}_{\mathrm{inl}(x)}) \qquad G_r(y) = \mathrm{inr}(y, \mathrm{refl}_{\mathrm{inr}(y)})$$

$$\text{By } \mathrm{ind}_+ : F : \prod_{z:A+B} P(z)$$

Chapter 1

Types and type-constructors

In this chapter we

1.1 Types vs Sets

All begun when Cantor's *Full Comprehension Scheme* (FCS):

$$\exists_u(u = \{x \mid \phi(x)\}),$$

where ϕ is any formula of $L = (\in)$, was proved contradictory for $\phi(x) := x \notin x$. Zermelo's *Restricted Comprehension Scheme* (RCS), also known as Separation Scheme,

$$\exists_u(u = \{x \in v \mid \phi(x)\})$$

replaced the FCS and it implies that $V \notin V$: if $V \in V$, then $u = \{x \in V \mid x \notin x\} \in V$ and then $u \in u \leftrightarrow u \notin u$. If FCS was not contradictory, we wouldn't need so many axioms to describe our intuition about sets. E.g., the union of two sets would be defined as $u \cup v = \{x \mid x \in u \vee x \in v\}$.

The first-order non-logical axioms of ZF in the first-order language $L = (\in)$ are the following:

Extensionality: $\forall_{x,y}(\forall_z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$.

Empty set: $\exists_x \forall_y(y \notin x)$.

Pair: $\forall_{x,y}\exists_z\forall_w(w \in z \leftrightarrow w = x \vee w = y)$.

Union: $\forall_x\exists_y\forall_z(z \in y \leftrightarrow \exists_w(w \in x \wedge z \in w))$.

Replacement Scheme: If $\phi(x, y, \vec{w})$ is a function formula, then

$$\forall_x\exists_v\forall_y(y \in v \leftrightarrow \exists_z(z \in x \wedge \phi(z, y, \vec{w})).$$

Power-set: $\forall_x\exists_y\forall_z(z \in y \leftrightarrow \forall_w(w \in z \rightarrow w \in x))$.

Foundation: $\forall_x(x \neq \emptyset \rightarrow \exists_z(z \in x \wedge \neg\exists_w(w \in z \wedge w \in x)))$.

Infinity: $\exists_x(\emptyset \in x \wedge \forall_y(y \in x \rightarrow y \cup \{y\} \in x))$.

Unlike group-axioms (first the models, the groups, and then the axioms) the set-axioms are given first and then we study their models!!! The axioms of ZF are generally "accepted" by standard mathematicians.

Definition 1.5.5

The axioms of ZF $(\mathcal{L} = (\in))$

1. Extensionality $\quad \forall_{x,y} \left(\forall_z (z \in x \leftrightarrow z \in y) \to x = y \right)$

2. Empty set $\quad \exists x \forall y (y \notin x)$

3. Pair $\quad \forall_{x,y} \exists_z \forall_w (w \in z \leftrightarrow w = x \vee w = y)$

4. Union $\quad \forall_x \exists_y \forall_z \left(z \in y \leftrightarrow \exists w (w \in x \wedge z \in w) \right)$

5. Replacement scheme $\quad \varphi(x, y, \bar{w})$ function formula

$$\forall_x \exists_v \forall_y \left(y \in v \to \exists_z (z \in x \wedge \varphi(z, y, \bar{w})) \right)$$

6. Power set $\quad \forall_x \exists_y \forall_z \left(z \in y \leftrightarrow \forall_w (w \in z \to w \in x) \right)$

7. Foundation $\quad \forall_x \left(x \neq \emptyset \to \exists_z (z \in x \wedge \underbrace{z \cap x = \emptyset}_{\neg \exists w (w \in z \wedge w \in x)}) \right)$

8. Infinity: $\quad \exists x \left(\emptyset \in x \wedge \forall y (y \in x \to y \cup \{y\} \in x) \right)$

ZFC $\equiv$ ZF + AC

AC: $\quad \forall_u \exists_{f : u \to V} \left(\forall_x (x \in u \to x \neq \emptyset \to f(x) \in x) \right)$

for every non-empty element x of u the choice function f selects an element of x. (eventually $f : u \to \bigcup u$)

AC_{∞} (XY): $\quad \forall_{x \in A} \exists_{y \in B} (R(x, y)) \to \exists_{f : A \to B} \forall_{x \in A} (R(x, f(x)))$

$AC \Rightarrow AC_{\infty}$: $\quad u_x = \{ y \in B \mid R(x, y) \}, \quad x \in A$

$u_x \in \mathcal{P}(B)$

$u = \{ u_x \mid x \in A \} = \{ z \in \mathcal{P}(B) \mid \exists x \in A \, (z = u_x) \}$

(AC): $f' : u \to \bigcup u \quad f'(u_x) \in u_x$ i.e., $R(x, f'(u_x))$.

Define $f(x) = f'(u_x)$, $x \in A$. (i.e. the set of pairs $(x, f'(u_x))$)

• ZFC + PEM, i.e. • GIVE THE AXIOMS of ZFC (in Bridges)

<u>Theorem</u> (Diaconescu, Goodman, Myhill) The (full) Axiom of Choice implies PEM
(in ZFC) • (1975) • (1978)

- In B. 67 p. 58 Ex. 2 as an exercise !!! (precursor of this result)
- In <u>Bridges</u>' thp another proof of it (independently from Diaconescu). (1967)

(Diaconescu)
<u>Proof</u>: P : Prop , $2 = \{0,1\}$

$A = \{x \in 2 \mid x = 0 \vee P\}$

$B = \{x \in 2 \mid x = 1 \vee P\}$

(In (CZF?) A, B are sets through full separation)
↳ what form of separation is there ??

• Classically, i.e. with PEM $(P \vee \neg P)$ $\forall P$: Prop

$A = \begin{cases} \{0,1\}, & \text{if } P \text{ is true} \\ \{0\}, & \text{if } P \text{ is false} \end{cases}$ (*)

$B = \begin{cases} \{0,1\}, & \text{if } P \text{ is true} \\ \{1\}, & \text{if } P \text{ is false} \end{cases}$

• Constructively we cannot even show that A, B are finite (bijected to a <u>specific</u> natural number)

• Take the set $\{A, B\}$

By **AC** there is a choice-function $f : \{A, B\} \to \bigcup\{A, B\}$ ($= A \cup B$)

i.e. $f(A) \in A$ and $f(B) \in B$ by def of A, B ↳

$f(A) = 0 \vee P$ and $f(B) = 1 \vee P$

$f(A), f(B) \in 2$

• if $f(A) = 1$, then P

• if $f(A) = 0$, then $f(B) = 0$, then P

$f(B) = 1$, then $\neg P$, since if P then $A = B$ (*)

$A = B \to f(A) = f(B)$ ↳ $0 = 1$ ↯

• The proof requires full separation □

• In Const. Set theory there is only predicative separation (bounded quantifiers) (Goodman, Myhill)
and the AC ⇒ a restricted form of PEM (still not acceptable constructively) [some proof]

• In MLTT or $HA^{<\omega}$ there is no separation at all (subsets of types are defined differently)
⇒ and the above argument cannot work. ✓

Bart case :

Section 1.10 The coproduct of types

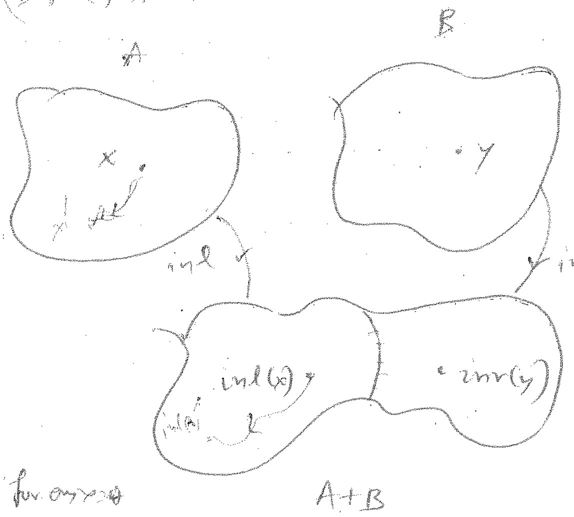
(cf. components & disjoint unions of sets)

(The line of presentation of types and type-constructors is such that the UP of each introduced type can be proven with the help of the previous material)

Definition 1.10.1

$$\mathrm{Form}_+ : \quad \frac{A, B : \mathcal{U}}{A + B : \mathcal{U}}$$

(injection left, — " — right) $\begin{pmatrix} a \mapsto (a, 1) \\ b \mapsto (b, 2) \end{pmatrix}$



$$\mathrm{Intro}_+ : \quad \frac{x : A}{\mathrm{inl}(x) : A+B} \qquad \frac{y : B}{\mathrm{inr}(y) : A+B}$$

$$\mathrm{Rec}_+ : \quad \frac{C : \mathcal{U}, \quad g_\ell : A \to C, \quad g_r : B \to C}{f : A+B \to C \quad \text{s.t.} \quad f(\mathrm{inl}(x)) \equiv g_\ell(x), \text{ for every } x : A}$$

$$f(\mathrm{inr}(y)) \equiv g_r(y), \quad \text{—"— } y : B$$

$$\mathrm{Ind}_+ : \quad \frac{\text{Let } P : A+B \to \mathcal{U} \qquad G_\ell : \prod_{x:A} P(\mathrm{inl}(x)), \quad G_r : \prod_{y:B} P(\mathrm{inr}(y))}{F : \prod_{z : A+B} P(z) \quad \text{s.t.} \quad F(\mathrm{inl}(x)) \equiv G_\ell(x), \text{ for every } x : A}$$

$$F(\mathrm{inr}(y)) \equiv G_r(y), \quad \text{—"— } y : B$$

• **Exercise** Formulate and prove the uniqueness principle for $+$

Prop. 1.10.2 (TFTH):
$$\prod_{z : A+B} \left(\sum_{x:A} (z =_{A+B} \mathrm{inl}(x)) + \sum_{y:B} (z =_{A+B} \mathrm{inr}(y)) \right)$$

Remark 1.10.3

• $x =_A x' \to \mathrm{inl}(x) =_{A+B} \mathrm{inl}(x')$ (by preservation of paths)

$y =_B y' \to \mathrm{inr}(y) =_{A+B} \mathrm{inr}(y')$

• What about the converse? (We cannot use path induction)
inl, inr injections → H. expected (shown later). (encode/decode)

~~Section 1.3 Dependent sum types (Σ types)~~

- Remark 1.10.4 By the FGP (intuitive) we expect $T \neq T\,11$

$$\prod_{x:A} \prod_{y:B} \left[\left(\mathrm{inl}(x) = \mathrm{inr}(y) \right) \to 0 \right]$$

This is FGP(+)

- We'll prove it later, independently from the decode – encode method. (Chapter 2).

Remark 1.2.10 : The Uniq. principle of some type expresses that every point of the type is (not ~~equal to~~ the output of some point-constructor, rather) equal to the output of some point-constructor. This will be the pattern for inductively defined types. The UP($\mathbb{Z}$) is a bit more delicate and will be considered later.

Section 1.11 : ~~The types boolean 2 and fin 2 naturals $\mathbb{N}$~~

Definition 1.11.1 The type of booleans 2.

AFTER COPRODUCT

Form$_2$: $2 : \mathcal{U}$

Intro$_2$: $\dfrac{}{0_2 : 2}$ $\dfrac{}{1_2 : 2}$

Rec$_2$: $\dfrac{A : \mathcal{U} \quad x : A \quad y : A}{f : 2 \to A \text{ st } f(0_2) \equiv x \text{ and } f(1_2) \equiv y}$

(A is "computation" type.)

Ind$_2$: $\dfrac{P : 2 \to \mathcal{U} \quad x : P(0_2) \quad y : P(1_2)}{F : \prod_{x:2} P(x) \text{ st } F(0_2) \equiv x \text{ and } F(1_2) \equiv y}$

Proposition 1.11.2 (Uniqueness Principle for 2) TFTI

(Exercise : Formulate UP(2) and prove it)

$$\prod_{x:2}\Big((x =_2 0_2) + (x =_2 1_2)\Big)$$

This will be shown after introducing coproducts.

Proof : $P(x) \equiv (x = 0_2) + (x = 1_2)$

$P(0_2)$, let $\mathrm{inl}(\mathrm{refl}_{0_2}) : P(0_2)$

$P(1_2)$, let $\mathrm{inr}(\mathrm{refl}_{1_2}) : P(1_2)$

now use Ind$_2$. ∎

Remark 1.11.3 FGP(2): there is $e_2 : (0_2 =_2 1_2) \to 0$.

Remark 1.11.? : We could have defined $2 \equiv 1+1$ (prove later that these are equivalent)

Definition 1.11.5 the type of naturals $\mathbb{N}$

- Form$_{\mathbb{N}}$: $\mathbb{N} : \mathcal{U}$

- Intr$_{\mathbb{N}}$: $\dfrac{}{0_{\mathbb{N}} : \mathbb{N}}$, $\dfrac{x : \mathbb{N}}{succ(x) : \mathbb{N}}$ (where $succ : \mathbb{N} \to \mathbb{N}$)

- Rec$_{\mathbb{N}}$:

$$\frac{A : \mathcal{U} \quad a_0 : A \quad s : \mathbb{N} \to A \to A}{f : \mathbb{N} \to A \text{ s.t. } \; f(0_{\mathbb{N}}) \equiv a_0, \quad f(succ(x)) \equiv s(x, f(x)), \text{ for any } x : \mathbb{N}}$$

(i.e. f is defined by prim. recursion)

- Ind$_{\mathbb{N}}$:

$$\frac{P : \mathbb{N} \to \mathcal{U}, \quad a_0 : P(0_{\mathbb{N}}), \quad S : \prod_{x : \mathbb{N}} (P(x) \to P(succ(x)))}{F : \prod_{x : \mathbb{N}} P(x) \text{ s.t. } \; F(0_{\mathbb{N}}) \equiv a_0, \quad F(succ(x)) \equiv S(x, F(x)), \text{ for any } x : \mathbb{N}}$$

(So we have a notion of standard induction on $\mathbb{N}$ which behaves the expected way.)

Prop 1.11.6 $Ind(\mathbb{N})$ $\vdash T$: $\prod_{x : \mathbb{N}} \Big((x =_{\mathbb{N}} 0_{\mathbb{N}}) + \underbrace{\sum_{y : \mathbb{N}} (x =_{\mathbb{N}} succ(y))}_{P(x)} \Big)$ (Ex)

Proof: $P(0) \equiv inl(refl_{0_{\mathbb{N}}})$

$S(x) : P(x) \to P(succ(x))$

$P(succ(x)) \equiv (succ(x) = 0_{\mathbb{N}}) + \sum_{y : \mathbb{N}} (succ(x) = succ(y))$, . . .

{ solution a new one 2η forgetting right }

Expect: $0 \neq succ(0)$ $x \neq succ(x)$ $\underline{FGA(\mathbb{N})}$

$succ(x) = succ(y) \to x = y$ (can't be one of inl, inr)

(60)

Proposition 1.11.8 If $x, y : \mathbb{N}$ such that $\boxed{succ(x) =_{\mathbb{N}} succ(y) \text{, then } x =_{\mathbb{N}} y.}$

(i.e. the type $succ(x) =_{\mathbb{N}} succ(y) \to x =_{\mathbb{N}} y$ is inhabited

Proof: By $Rec_{\mathbb{N}}$ we define $pred : \mathbb{N} \to \mathbb{N}$ as follows

$$0_{\mathbb{N}} : \mathbb{N}$$

$$s : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \quad \text{s.t.} \quad s(x, y) \equiv x \quad \text{, for every } x, y : \mathbb{N}$$

Then there is $pred : \mathbb{N} \to \mathbb{N}$ s.t. $pred(0_{\mathbb{N}}) \equiv 0_{\mathbb{N}}$

$$- \; pred(succ(x)) \equiv s(x, pred(x)) \equiv x$$

By preservation of paths

$$ap_{pred} : \left(succ(x) =_{\mathbb{N}} succ(y) \right) \to pred(succ(x)) =_{\mathbb{N}} pred(succ(y))$$

i.e.,

$$ap_{pred} : \left(succ(x) =_{\mathbb{N}} succ(y) \right) \to \left(x =_{\mathbb{N}} y \right)$$

□

(Much simpler than ind, in case of injectivity).

and simpler than § 2.13 in HTT-book.

Section 2.1 On some inequalities

Proposition 2.1.1: The following types are inhabited

(a) $0_2 \neq_2 1_2 \leftrightarrow 0_{\mathbb{N}} \neq_{\mathbb{N}} 1_{\mathbb{N}}$

(b) $0_2 \neq_2 1_2 \to \mathrm{inl}(a) \neq \mathrm{inr}(b)$ $a:A$ $b:B$

($\leftarrow$ this is also the case but requires the proof of $2 \simeq 1+1$)
i.e. it is based on the notion of equivalence.

Proof: (a) $0_2 \neq_2 1_2 \to 0_{\mathbb{N}} \neq 1_{\mathbb{N}}$:

Let $r : \mathbb{N} \to 2$ defined by $\mathrm{Rec}_{\mathbb{N}}$: $0_2 : 2$

$s : \mathbb{N} \to 2 \to 2$ is defined by

$$s \equiv \lambda(x:\mathbb{N}).\,(\lambda(y:2).\,1_2)$$

$\underline{x \mapsto s(x)}$ = the constant 1_2 on 2

i.e. $s \equiv \overline{(1_2)}_{2\,\mathbb{N}}$

By $\mathrm{Rec}_{\mathbb{N}}$ $r : \mathbb{N} \to 2$ s.t. $r(0_{\mathbb{N}}) \equiv 0_2$

$r(\mathrm{succ}(x)) \equiv s(x, r(x)) \equiv 1_2$

Let $p : 0_{\mathbb{N}} = 1_{\mathbb{N}}$

$$\mathrm{ap}_r : (0_{\mathbb{N}} = 1_{\mathbb{N}}) \to \left(r(0_{\mathbb{N}}) =_2 r(1_{\mathbb{N}})\right) \equiv (0_2 =_2 1_2) \xrightarrow{e} 0$$

$$e \circ \mathrm{ap}_r : (0_{\mathbb{N}} =_{\mathbb{N}} 1_{\mathbb{N}}) \to 0.$$

$0_{\mathbb{N}} \neq 1_{\mathbb{N}} \to 0_2 \neq 1_2$: By Rec_2 There is $t : 2 \to \mathbb{N}$: $t(0_2) \equiv 0_{\mathbb{N}}$
$t(1_2) \equiv 1_{\mathbb{N}}$

$$\mathrm{ap}_t : 0_2 =_2 1_2 \to (0_{\mathbb{N}} =_{\mathbb{N}} 1_{\mathbb{N}}) \xrightarrow{e'} 0$$

(b) ~~[illegible]~~ Let $i : A \to 2$ $i \equiv \overline{(0_2)}_A$

$j : B \to 2$ $j \equiv \overline{(1_2)}_B$

By Rec_+ there is $f : A+B \to 2$ s.t. $f(\mathrm{inl}(a)) \equiv i(a) \equiv 0_2$
$f(\mathrm{inr}(b)) \equiv j(b) \equiv 1_2$

$$\mathrm{ap}_f : \left(\mathrm{inl}(a) =_{A+B} \mathrm{inr}(b)\right) \to (0_2 =_2 1_2) \xrightarrow{e} 0$$

$\square$

Proposition 21.5 The following type is inhabited, if $A, B : \mathcal{U}$, $a : A$

$$\prod_{z:A+B}\ \prod_{p:inl(a)=z} \left(\sum_{x:A} (z =_{A+B} inl(x)) \right)$$

(or $inl(x) =_{A+B} z$)

i.e., if $inl(a) = z$, then $z = inl(x)$, for some $x : A$. (argument for inr)

Proof 1 (b-Ind): Let $C : \prod_{z:A+B}\ \prod_{p:inl(a)=z} \mathcal{U}$ be defined by

$$C(z, p) \equiv \sum_{x:A} (z =_{A+B} inl(x))$$

here $C(inl(a), refl_{inl(a)}) \equiv \sum_{x:A} (inl(a) =_{A+B} inl(x))$

let $c \equiv (a, refl_{inl(a)}) : C(inl(a), refl_{inl(a)})$

By b-Ind$_2$ there is $F : \prod_{z:A+B}\ \prod_{p:inl(a)=z} \left(\sum_{x:A} (z =_{A+B} inl(x)) \right)$

s.t. $F(inl(a), refl_{inl(a)}) \equiv (a, refl_{inl(a)})$

□

Let $E_{inl(a)} \equiv \sum_{z:A+B} \underbrace{(inl(a) =_{A+B} z)}_{P(z)}$

Corollary TFT!! $\prod_{u:E_{inl(a)}}\ \sum_{x:A} (pr_1(u) =_{A+B} inl(x))$

(or $(inl(x) =_{A+B} pr_1(u))$)

$pr_1 : E_{inl(a)} \to A+B$, i.e. if $u : E_{inl(a)}$

$pr_1(u) : A$

Proof: $pr_2(u) : (inl(a) =_{A+B} pr_1(u))$

We define $G(u) \equiv F(pr_1(u), pr_2(u)) : \sum_{x:A} (pr_1(u) =_{A+B} inl(x))$.

Clearly $G \equiv \lambda(u : E_{inl(a)}).\, G(u)$ is the required inhabitant.

- Proposition 2.1.2 $\quad 2 : \mathcal{U} \quad 2' : \mathcal{U}' \quad 0', 1' : 2'$.

$$0' =_2 1' \to 0 =_2 1 \quad \text{is inhabited}$$

Proof: By Rec$_{2'}$ there is $f : 2' \to 2$ s.t. $f(0') \equiv 0$ $f(1') \equiv 1$

$\mathrm{ap}_f : (0' =_{2'} 1') \to (0 =_2 1)$ is the required inhabitant.

- Corollary 2.1.3: $0 \neq_2 1 \to ((0' =_2 1') \to 0)$

Proof: By $(A \to B) \to (\neg B \to \neg A) \equiv (A \to B) \to ((B \to 0) \to (A \to 0))$

and Prop. 2.1.2.

if $f : A \to B$ then $\hat{f} : (B \to 0) \to (A \to 0)$ is defined by

$\hat{f}(u) : A \to 0 \quad , u : B \to 0$

$\hat{f}(u)(x) = u(f(x)) \quad , x : A$

- One of the two next facts can be used in the proof of the Theorem 2.1.4.

(a) There is $g : 2 \to 2'$ s.t. $g(0_2) \equiv 0'_{2'}$, $g(1_2) \equiv 1'_{2'}$

without this there is only the constant $0'$, and constant $1'$ that can be defined $2 \to 2'$ using λ-abstraction. This fact (a) is accepted in Agda.

(b) If $p : 0_2 =_2 1_2$, then there is $p' : 0' =_{2'} 1'$

Since 2 and $2'$ behave the same in $\mathcal{U}$ and $\mathcal{U}'$, then if there was some p one would expect some p'.

Actually we'd like to have ~~...~~ $t : (0 =_2 1) \to (0' =_{2'} 1')$ while $(0 = 1) \to (0' = 1')$ is in $\mathcal{U}'$.

(a) $\Rightarrow$ (b): $\quad \mathrm{ap}_g : (0_2 =_2 1_2) \to (0'_2 =_2 1'_2)$.

$p \mapsto \mathrm{ap}_g(p)$

Theorem 2.1.4 (+ (a) or (b)) The type $(0_2 =_2 1_2) \to 0$ is inhabited

Proof: Let $P : 2' \to \mathcal{U}'$ defined by

$P(x') \equiv (0'_{2'} =_{2'} x') \to \mathcal{U} \equiv \prod_{p' : 0_2 =_{2'} x'} \mathcal{U}$, note that $0'_2 =_{2'} x'$ and $\mathcal{U}$ are types in $\mathcal{U}'$.

$P(0') \equiv (0'_{2'} =_{2'} 0'_{2'}) \to \mathcal{U}$

$P(1') \equiv (0'_{2'} =_{2'} 1') \to \mathcal{U}$.

$$u \equiv \lambda(p : 0'_{2'} =_{2'} 0'_{1'}) . \underbrace{1}_{\text{constant maps}} : P(0')$$

$$w \equiv \lambda(p : 0'_{2'} =_{2'} 1'_{2'}) . 0 \quad : P(1')$$

[either $0_{2'}, 1_{2'}$ or $0', 1'$]

By $\mathrm{Ind}_{2'}$ there is $C : \prod_{x' : 2} \prod_{p' : 0' = x'} \mathcal{U}$ $\quad C(0') \equiv u, \; C(1') \equiv w$

since $C(x', p') : \mathcal{U}$ we also get

$$\frac{}{C(x', p') : \mathcal{U}'} \qquad \left(\frac{A : \mathcal{U}}{A : \mathcal{U}'}\right)$$

i.e., $C : \prod_{x' : 2} \prod_{p' : 0' = x'} \mathcal{U}'$ s.t. $C(0', p) \equiv 1$

$C(1', p) \equiv 0$

and $0_1 : C(0', \mathrm{refl}_{0'}) \equiv 1$

By based-$\mathrm{Ind}_=$ in $\mathcal{U}'$, there is $F : \prod_{x' : 2} \prod_{p' : 0' = x'} C(x', p')$

s.t. $F(0', \mathrm{refl}_{0'}) = 0_1$.

Moreover $F(1', p') : C(1', p') \equiv 0$.

We define $e : (0 =_2 1) \to 0$ by

$$e(p) \equiv F(1', \mathrm{ap}_g(p)) \quad , \text{for every } p : 0 =_2 1$$

$\square$

Note: In the def of $A \to B$ there is no restriction to the $b(x) : B$, $b(x)$ can be any rule, here we need F, $1' : 2'$

Remark: This proof should be implemented in Agda (Xu)

There is another simple proof of $0 \neq 1$ with pattern matching (Coquand).

The proof of Proposition gives us no information on the value $F(\mathrm{inl}(x),\ p : \mathrm{inl}(a) =_{A+B} \mathrm{inl}(x))$, which is expected to be $(x, \mathrm{refl}_{\mathrm{inl}(x)})$. So, we give a more informative proof of this proposition (the corollary remains as it is).

PROOF of Prop 2.1.5

Proof 2:

We define
$$F : \prod_{z:A+B} P(z) \equiv \prod_{z:A+B} \underbrace{\left(\prod_{p:\mathrm{inl}(a)=_{A+B} z} \left(\sum_{x:A} (z =_{A+B} \mathrm{inl}(x))\right)\right)}_{P(z)}$$

by ind_+.

We define
$$G_\ell : \prod_{x:A} P(\mathrm{inl}(x)) \equiv \prod_{x:A} \left(\prod_{p:\mathrm{inl}(a)=_{A+B}\mathrm{inl}(x)} \left(\sum_{x':A} (\mathrm{inl}(x) =_{A+B} \mathrm{inl}(x'))\right)\right)$$

by $G_\ell(x, p) \equiv (x, \mathrm{refl}_{\mathrm{inl}(x)})$

We define
$$G_r : \prod_{y:B} P(\mathrm{inr}(y)) \equiv \prod_{y:B} \left(\prod_{p:\mathrm{inl}(a)=_{A+B}\mathrm{inr}(y)} \left(\sum_{x:A} \underbrace{(\mathrm{inr}(y) =_{A+B} \mathrm{inl}(x))}\right)\right)$$

by $G_r(y, p) \equiv \mathrm{Efq}_{Q(p)}(e(p)) : Q(p)$

where if $e : \mathrm{inl}(a) =_{A+B} \mathrm{inr}(y) \to 0$, already established, and

$$\mathrm{Efq}_{Q(p)} : 0 \to Q(p)$$

Hence by ind_+ $F : \prod_{z:A+B} P(z)$ s.t. $F(\mathrm{inl}(x)) \equiv G_\ell(x)$
$F(\mathrm{inr}(y)) \equiv G_r(y)$

Hence $F(\mathrm{inl}(x), \overset{p}{\cancel{\mathrm{refl}_{\mathrm{inl}(x)}}}) \equiv (x, \mathrm{refl}_{\mathrm{inl}(x)})$ and as a special case $x \equiv a$: we get
$F(\mathrm{inl}(a), \cancel{\mathrm{refl}_{\mathrm{inl}(a)}}) \equiv (a, \mathrm{refl}_{\mathrm{inl}(a)})$ $\square$

More importantly:

Proposition 2.1.5 $A, B : \mathcal{U}$, $a : A$ TFTII $\Rightarrow$

$$\prod_{z : A+B} \prod_{p : inl(a) = z} \left(\sum_{x : A} (z =_{A+B} inl(x)) \right)$$

by some F such that

$$F(inl(x), p : inl(a) = inl(x)) \equiv (x, refl_{inl(x)}), \quad \text{for every } x : A.$$

Corollary 2.1.6 Let $E_{inl(a)} \equiv \sum_{z : A+B} (inl(a) =_{A+B} z)$.

There is $\Phi : \prod_{u : E_{inl(a)}} \sum_{x : A} (pr_1(u) =_{A+B} inl(x))$ s.t. $\Phi((z,p)) \equiv F(z,p)$

where F from Prop. 2.1.1

Proof. By ind_{Σ} : let $Q : E_{inl(a)} \to \mathcal{U}$

$$Q(u) \equiv \sum_{x : A} (pr_1(u) =_{A+B} inl(x))$$

We define $G : \prod_{z : A+B} \prod_{p : inl(a) =_{A+B} z} Q((z,p)) \equiv \prod_{z : A+B} \prod_{p : inl(a) =_{A+B} z} \sum_{x : A} (z =_{A+B} inl(x))$

Let $G(z,p) \equiv F(z,p)$, where F is defined in Proposition 2.1.5.

Hence there is $\Phi : \prod_{u : E_{inl(a)}} Q(u)$ s.t. $\Phi((z,p)) \equiv F(z,p)$. ∎

Theorem 2.1.7 $A, B : \mathcal{U}$, $x : A$, $a : A$. TFTII: $inl(x) =_{A+B} inl(a) \to x =_A a$

Proof. By the proof of the axiom of choice we have for

$$ac(\Phi) : \sum_{f : E_{inl(a)} \to A} \prod_{u : E_{inl(a)}} (pr_1(u) =_{A+B} inl(f(u)))$$

and $f \equiv \lambda(u : E_{inl(a)}).\ pr_1(\Phi(u))$, hence if $x : A$,

$$f((inl(x), p : inl(a) =_{A+B} inl(x))) \equiv pr_1(\Phi((inl(x), p)))$$

$$\overset{2.1.6}{\equiv} pr_1(F(inl(x), p))$$

$$\overset{2.1.5}{\equiv} pr_1(x, refl_{inl(x)}) \equiv x$$

For $x \equiv a$ we get $f((\mathrm{inl}(a), \mathrm{refl}_{\mathrm{inl}(a)})) \equiv a$

By $\mathrm{ut}(=)$, if $u : E_{\mathrm{inl}(a)}$ then

$$\Lambda_{\mathrm{inl}(a)}(u) : u =_{E_{\mathrm{inl}(a)}} (\mathrm{inl}(a), \mathrm{refl}_{\mathrm{inl}(a)})$$

hence for $u \equiv (\mathrm{inl}(x), p)$

$$\Lambda_{\mathrm{inl}(a)}((\mathrm{inl}(x), p)) : (\mathrm{inl}(x), p) =_{E_{\mathrm{inl}(a)}} (\mathrm{inl}(a), \mathrm{refl}_{\mathrm{inl}(a)})$$

By the preservation of paths for f:

$$\mathrm{ap}_f\left(\Lambda_{\mathrm{inl}(a)}((\mathrm{inl}(x), p))\right) : f((\mathrm{inl}(x), p)) =_A f((\mathrm{inl}(a), \mathrm{refl}_{\mathrm{inl}(a)}))$$

$$\begin{array}{ccc} f((\mathrm{inl}(x), p)) & =_A & f((\mathrm{inl}(a), \mathrm{refl}_{\mathrm{inl}(a)})) \\ \equiv & & \equiv \\ x & =_A & a \end{array}$$

$\square$